Poster : Blockchain abstract data type

Emmanuelle Anceaume, Antonella del Pozzo, Romaric Ludinard, Maria Potop-Butucaru, Sara Tucci-Piergiovanni

POSTER: Blockchain Abstract Data Type

Emmanuelle Anceaume
CNRS, IRISA
France

Antonella Del Pozzo
Institut LIST, CEA, Université
Paris-Saclay, F-91120, Palaiseau
France

Romaric Ludinard
IMT Atlantique, IRISA
France

Maria Potop-Butucaru
Sorbonne Université, CNRS,
Laboratoire d'Informatique de Paris 6
France

Sara Tucci-Piergiovanni
Institut LIST, CEA, Université
Paris-Saclay, F-91120, Palaiseau
France

Abstract

This paper is the first to specify blockchains as a composition of *abstract data types* all together with a hierarchy of *consistency criteria* that formally characterizes the histories admissible for distributed programs that use them. The paper presents as well some results on implementability of the presented abstractions and a mapping of representative existing blockchains from both academia and industry in our framework.

CCS Concepts • **Theory of computation → Distributed computing models**;

Keywords Blockchain, Abstract Data Type, Consistency Criteria

1 Introduction

The paper proposes a new data type to formally model blockchains and their behaviors. We aim at providing consistency criteria to capture the correct behavior of current blockchain proposals in a unified framework encompassing both forkable and non-forkable blockchains. To this end, the key point is to define consistency criteria allowing mutable operations to create forks and restricting the values read, i.e. modeling the data structure as an append-only tree. This way we can easily define semantics equivalent to eventually consistent append-only queue for non-forkable blockchains but as well as weaker semantics for forkable ones. For non-forkable blockchain we introduce the Strong Prefix consistency criterion that guarantees that for any two reads, the value returned by one is a prefix of the value returned by the other one. A similar proposal has been presented in [10]. For forkable blockchains, we introduce the Eventual Prefix consistency criterion, which guarantees that eventually, for any two reads, the value returned by one read is a prefix of the value returned by the other one.

Another peculiarity of blockchains lies in the notion of *validity* of blocks, i.e. the blockchain must contain only blocks that satisfy a given predicate. To abstract away implementation-specific validation mechanisms, we propose to encapsulate the validation process in an oracle model separated from the process of updating the data structure. The oracle has two roles: first generating valid blocks, to be potentially appended to the append-only tree data structure, and second managing the append of the new valid blocks. This second role allows forks in the append-only tree data structure: from an unbounded number of forks (weak oracle) to none of them (strong oracle). The blockchain is then abstracted by an oracle-based construction in which the update and consistency of the tree data structure depend on the validation and synchronization power of the oracle.

Related Work. In [8] the authors extract Bitcoin backbone and define invariants that this protocol has to satisfy in order to verify with high probability an eventual consistent prefix. This line of work has been continued by [14]. However, to the best of our knowledge, no other previous attempt proposed a consistency unified framework and hierarchy capturing both Consensus-based and proof-of-work based blockchains. In [1], the authors present a study about the relationship of BFT consensus and blockchains. In order to abstract out the proof-of-work mechanism the authors propose a specific oracle, in the same spirit of our oracle abstraction, but more specific than ours, since it makes a direct reference to proof-of-work properties. In parallel and independently of our work, [4] proposes a formalization of distributed ledgers modeled as an ordered list of records. The authors propose in their formalization three consistency criteria: eventual consistency, sequential consistency and linearizability. Interestingly, they show that a distributed ledger that provides eventual consistency can be used to solve the consensus problem. These findings confirm our results about the necessity of Consensus to solve Strong Prefix. On the other hand, the proposed formalization does not propose weaker consistency semantics more suitable for proof-of-work blockchains

PPoPP '19, February 16–20, 2019, Washington, DC, USA

ACM ISBN 978-1-4503-6225-2/19/02.
https://doi.org/10.1145/3293883.3303705

as BitCoin. The work achieved in [4] is complementary to the one presented in [2], where the authors study the consistency of blockchain by modeling it as a register. Finally, [10] presents an implementation of the Monotonic Prefix Consistency (MPC) criterion and showed that no criterion stronger than MPC can be implemented in a partition-prone message-passing system.

2 Contribution

The main contribution of the paper is a formal unified framework providing blockchain consistency criteria that can be combined with oracle models in a proper *hierachy of abstract data types* [15] independent of the underlying communication and failure model. The following implementability results are shown:

- The strongest oracle, guaranteeing no fork, has Consensus number ∞ in the Consensus hierarchy of concurrent objects [11]. It must be noted that we considered Consensus defined in [5, 6, 9], in which the *Validity* property states that a valid block can be decided even if sent by a faulty process.
- The weakest oracle, which validates a potentially unbounded number of blocks to be appended to a given block, has Consensus number 1.
- The impossibility to guarantee Strong Prefix in a message-passing system if forks are allowed. This means that Strong Prefix needs the strongest oracle to be implemented, which is at least as strong as Consensus.
- A necessary condition for Eventual Prefix in a message-passing system, called *Update Agreement* stating that each update sent by a correct process must be eventually received by every correct process. Thus, it is impossible to implement Eventual Prefix if even only one message sent by a correct process is dropped.

The proposed framework along with the above-mentioned results helps in classifying existing blockchains in terms of their consistency and implementability. We used the framework to classify several blockchain proposals. We showed that Bitcoin [13] and Ethereum [16] have a validation mechanism that maps to our weakest oracle and then they only implement Eventual prefix, while other proposals map to our strongest oracle, falling in the class of those that guarantee Strong Prefix (e.g. Hyperledger Fabric [3], PeerCensus [7], ByzCoin [12]).

3 Conclusion and Future Work

We believe that the presented results are of practical interests since our oracle construction not only reflects the design of many current implementations but will help designers in choosing the oracle they want to implement with a clear semantics and inherent trade-offs in mind. Future work will focus on several open issues, such as the solvability of Eventual Prefix in message-passing, the synchronization power of other oracle models, and fairness properties for oracles.

References

[1] Ittai Abraham and Dahlia Malkhi. 2017. The Blockchain Consensus Layer and BFT. *Bulletin of the EATCS* 123 (2017).

[2] Emmanuelle Anceaume, Romaric Ludinard, Maria Potop-Butucaru, and Frédéric Tronel. 2017. Bitcoin a Distributed Shared Register. In *Proceedings of the 19th International Symposium Stabilization, Safety, and Security of Distributed Systems (SSS 2017)*.

[3] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. VukoliÄĞ, S. Weed Cocco, and J. Yellick. 2018. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. https://arxiv.org/pdf/1801.10228v1.pdf.

[4] A. Fernández Anta, C. Georgiou, K. M. Konwar, and N. C. Nicolaou. 2018. Formalizing and Implementing Distributed Ledger Objects. *CoRR* abs/1802.07817 (2018). http://arxiv.org/abs/1802.07817

[5] C. Cachin, K. Kursawe, F. Petzold, and V. Shoup. 2001. Secure and Efficient Asynchronous Broadcast Protocols. In *Proceedings of Advances in Cryptology - CRYPTO, 21st Annual International Cryptology Conference*.

[6] T. Crain, V. Gramoli, M. Larrea, and M. Raynal. 2017. (Leader/Randomization/Signature)-free Byzantine Consensus for Consortium Blockchains. http://arxiv.org/abs/1702.03068.

[7] C. Decker, J. Seidel, and R. Wattenhofer. 2016. Bitcoin Meets Strong Consistency. In *Proceedings of the 17th International Conference on Distributed Computing and Networking Conference (ICDCN)*.

[8] J. A. Garay, A. Kiayias, and N. Leonardos. 2015. The Bitcoin Backbone Protocol: Analysis and Applications. In *Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*.

[9] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 51–68.

[10] Alain Girault, Gregor Gössler, Rachid Guerraoui, Jad Hamza, and Dragos-Adrian Seredinschi. 2018. Monotonic Prefix Consistency in Distributed Systems. In *International Conference on Formal Techniques for Distributed Objects, Components, and Systems*. Springer, 41–57.

[11] Maurice Herlihy. 1991. Wait-free synchronization. *ACM Transactions on Programming Languages and Systems (TOPLAS)* (1991).

[12] E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford. 2016. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In *Proceedings of the 25th USENIX Security Symposium*.

[13] S. Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf.

[14] Rafael Pass and Elaine Shi. 2017. FruitChains: A Fair Blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017*.

[15] M. Perrin, A. Mostefaoui, and C. Jard. 2016. Causal Consistency: Beyond Memory. In *21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*.

[16] G. Wood. 2014. Ethereum: A secure decentralised generalised transaction ledger. http://gavwood.com/Paper.pdf.